

CSc 174

Database Management Systems

14. Recovery Techniques

Ying Jin

Computer Science Department

California state University, Sacramento

Purpose of Database Recovery

- ◆ Database is restored to the most recent consistent state

- ◆ Preserve ACID properties

- ◆ Example:

Transfer (withdraw+deposit).

If part of the transaction fails, the account balances should not be changed

Checkpoints

- ◆ [commit,T] entries in the log: T committed
- ◆ Checkpoint: Indicates that the system has written the effect of all write operations of **committed** transactions to the disk .
- ◆ The updates from all committed transactions before [checkpoint] are recorded on disk during checkpointing.

Recovery Techniques

- ◆ Two main techniques for recovery:
 - Deferred update
 - Immediate update
- ◆ Redo/Undo

Deferred update technique

- ◆ The deferred update techniques do not physically update the database on disk until *after* a transaction reaches its commit point.
- ◆ During transaction execution, the updates are recorded only in the log and in the cache buffers.

Deferred Update protocol

◆ Protocol

1. A transaction cannot change the database on disk until it reaches its commit point
2. A transaction does not reach its commit point until all its update operations are recorded in the log and the log is force-written to disk.

UNDO on Deferred Update technique

◆ No UNDO

- Because the database never records updates on disk before a transaction commit, there is never a need to UNDO any operation.
- Log entries for uncommitted transactions are ignored because writes are deferred

REDO on Deferred Update technique

- ◆ REDO is needed in case the system fails after a transaction commits but before all its changes are recorded in the database on disk
- ◆ If a transaction commits before a checkpoint, no need to redo, since the updates have been recorded on disk
- ◆ If a transaction's commit point is after the last system checkpoint, it needs to be redone.

Deferred Update technique

-Example of redo

[start-trans T1]

...

[commit T1]

[start-trans T2]

....

A:

[start-trans T3]

...

[checkpoint]

B:

....

[commit T2]

C:

◆ Failed at A, B, C

◆ A: redo T1

◆ B: do nothing

◆ C:redo T2

Immediate update technique

- ◆ The database may be updated by some operations of a transaction before the transaction reaches its commit point.
- ◆ These operations are recorded in the log on disk by force writing before they are applied to the database, making recovery possible.

UNDO/REDO on Immediate Update technique

- ◆ In case of failure, **undo** all the write_item() operations on **uncommitted** transaction.
- ◆ Redo: committed transaction
 - If a transaction's commit point is after the last system checkpoint, it needs to be redone.
 - Reason: A transaction commit, but its updates may not be recorded in the database on disk.
- ◆ If a transaction commits before a checkpoint, no need to redo/undo, since the updates have been recorded in the database on disk.

Immediate Update technique

-Example of redo/undo

[start-trans T1]

...

[commit T1]

[start-trans T2]

....

A:

[start-trans T3]

...

[checkpoint]

B:

....

[commit T2]

C:

◆ Failed at A, B, C

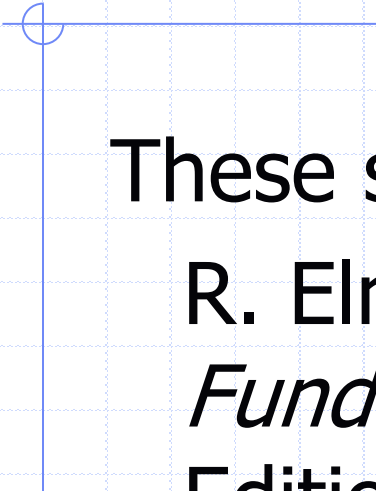
◆ A: redo T1, undo T2

◆ B: undo T2, T3

◆ C:redo T2, undo T3

Redo/Undo

- ◆ Redo & Undo must be idempotent – “executing it several times is equivalent to executing it once.”
- ◆ Redo all the write operations of the committed transactions from the log, in the order in which they were written into the log
- ◆ Undo operations from the log must proceed in the *reverse* order from the order in which the operations were written in the log.



These slides are based on the textbook:
R. Elmasri and S. Navathe,
Fundamentals of Database System, 7th
Edition, Addison-Wesley.